US Patent 5,611,049, Claim 1:

1. In **a network of digital computers** that includes **a first plurality of Network Distributed Cache ("NDC") sites**, each NDC site including **an NDC** that has **an NDC buffer, a method for projecting images of a stored dataset** from **an NDC server terminator site** into **a second plurality of NDC client terminator sites** in response to **requests to concurrently access such stored dataset** transmitted from **a third plurality of client sites** respectively to the **second plurality of NDC client terminator sites**, the method comprising the steps of:

(a) **the NDC receiving the request** to access data in the stored dataset;

(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there;

(c) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, **the NDC of this NDC site transmitting a request for data** from this NDC site **downstream to another NDC site** closer to **the NDC server terminator site** for the stored dataset than the present NDC site;

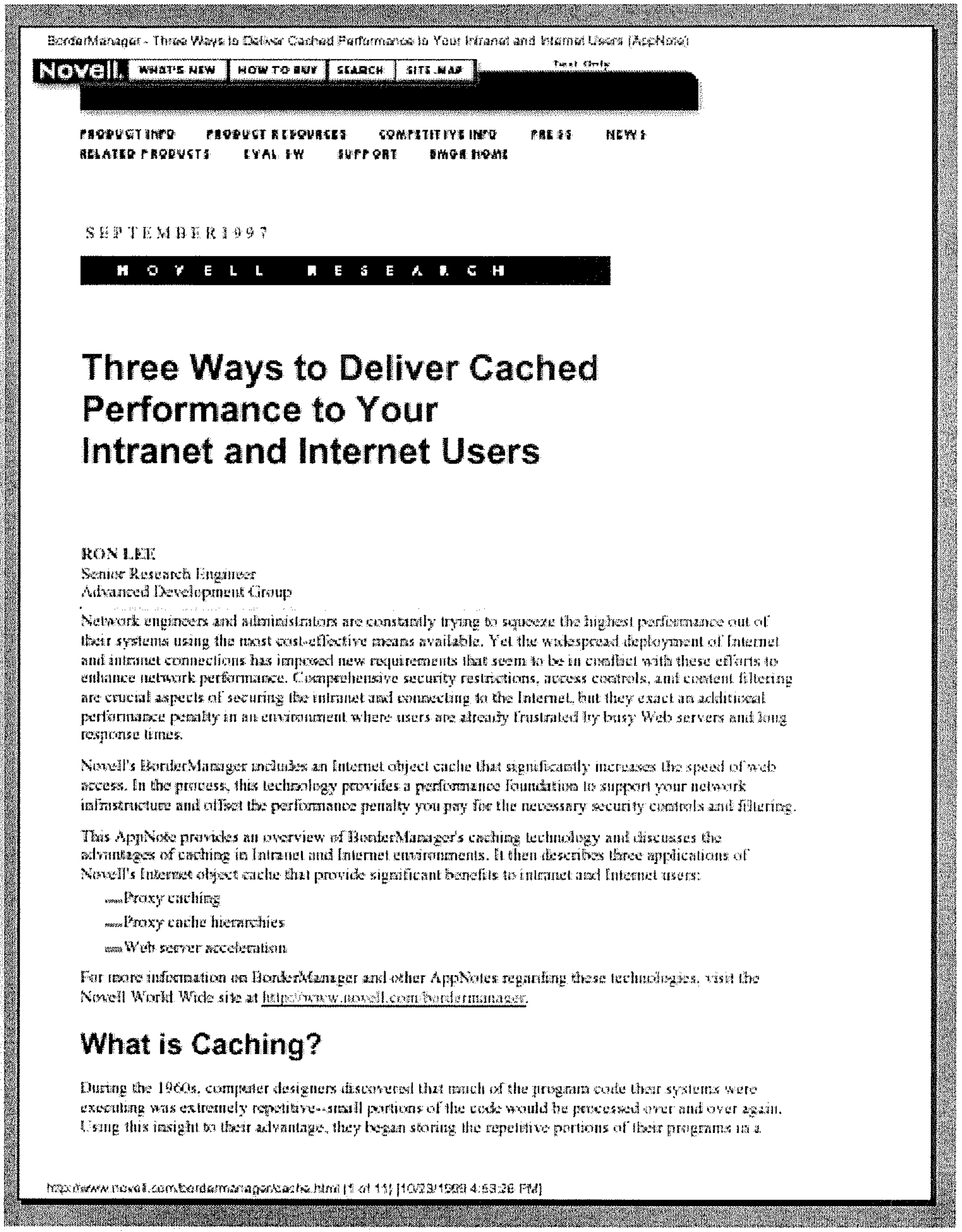
(d) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is the NDC server terminator site** for the stored dataset, **the NDC of the NDC server terminator site accessing the stored dataset** to **project an image of the requested data** into the NDC buffer of the NDC server terminator site;

(e) **repeating the steps (a) through (d)** until the NDC buffer of the **downstream NDC site receiving the request** contains **a projected image of all requested data;**

(f) **each successive NDC site**, having obtained **a projected image of all the requested data, returning the requested data** upstream to the NDC site from which the NDC site received the request until **the requested data arrives at the NDC client terminator site**, each NDC site that returns data upstream to the requesting NDC site **retaining a copy of the returned data** that **the returning NDC site** may **subsequently transmit to an NDC site other than the NDC site to which the returning NDC site first returned the data**, whereby images of the stored dataset may be projected concurrently from **a single NDC site** into **the second plurality of NDC client terminator sites**; and

(g) **the NDC client terminator site**, upon **receiving the requested data, returning the requested data to the client site** that requested access to the stored dataset.

049.1
2



BorderManager - Three Ways to Deliver Cached Performance to Your Intranet and Internet Users (AppNote)

SEPTEMBER 1997

NOVELL RESEARCH

Three Ways to Deliver Cached Performance to Your Intranet and Internet Users

RON LEE
Senior Research Engineer
Advanced Development Group

Network engineers and administrators are constantly trying to squeeze the highest performance out of their systems using the most cost-effective means available. Yet the widespread deployment of Internet and intranet connections has imposed new requirements that seem to be in conflict with these efforts to enhance network performance. Comprehensive security restrictions, access controls, and content filtering are crucial aspects of securing the intranet and connecting to the Internet, but they exact an additional performance penalty in an environment where users are already frustrated by busy Web servers and long response times.

Novell's BorderManager includes an Internet object cache that significantly increases the speed of web access. In the process, this technology provides a performance foundation to support your network infrastructure and offset the performance penalty you pay for the necessary security controls and filtering.

This AppNote provides an overview of BorderManager's caching technology and discusses the advantages of caching in Intranet and Internet environments. It then describes three applications of Novell's Internet object cache that provide significant benefits to intranet and Internet users:

- Proxy caching
- Proxy cache hierarchies
- Web server acceleration

For more information on BorderManager and other AppNotes regarding these technologies, visit the Novell World Wide site at http://www.novell.com/bordermanager.

What is Caching?

During the 1960s, computer designers discovered that much of the program code their systems were executing was extremely repetitive--small portions of the code would be processed over and over again. Using this insight to their advantage, they began storing the repetitive portions of their programs in a

http://www.novell.com/bordermanager/cache.html (1 of 11) [10/23/1998 4:53:26 PM]

(a) **the NDC receiving the request** to access data in the stored dataset; ✓

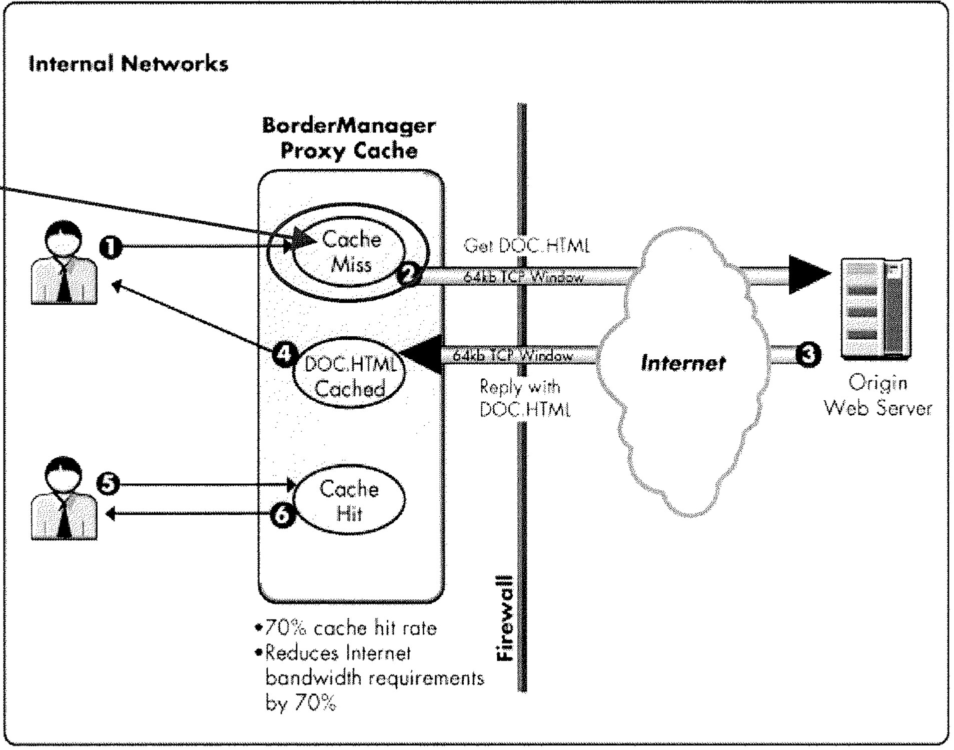
(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there; ✓

(c) if the NDC buffer of this NDC site does not contain a projected image of all data requested from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, **the NDC of this NDC site transmitting a request for data** from this NDC site **downstream to another NDC site** closer to **the NDC server terminator site** for the stored dataset than the present NDC site;

How Proxy Cache Works

Figure 5 illustrates how BorderManager caches HTML documents and other cacheable content.

Figure 5: A proxy cache saves repeatedly-used objects to speed access and reduce Internet traffic.



1. A browser issues a request for a file named DOC.HTML. This request is sent to the proxy cache over a 10 Mbps Ethernet LAN segment. In this case, the request results in a "cache miss" because the proxy cache has never serviced a request for that document before.
2. The proxy cache initiates a request for DOC.HTML from the origin web server on behalf of the browser. This request is sent over a T1 line to an ISP, then traverses the Internet until it arrives at the origin server.
3. The origin web server responds to the proxy's request by sending DOC.HTML. This transmission is much faster than a response to a browser due to the proxy's optimized receive window that can receive up to 64KB at one time and stays open to receive multiple responses. The proxy then places DOC.HTML in its cache.
4. The proxy cache responds to the original browser request with DOC.HTML.
5. Now when the same browser (or any other browser) issues a request for DOC.HTML, the request results in a "cache hit" because the proxy has kept a copy of the document in its cache.
6. In this case, the proxy replies immediately to the browser request because it has DOC.HTML in cache. The proxy's response is transmitted at 10 Mbps to the browser, eliminating the need to fetch the document again from the origin server on the Internet.

049.1
19

(a) **the NDC receiving the request** to access data in the stored dataset;

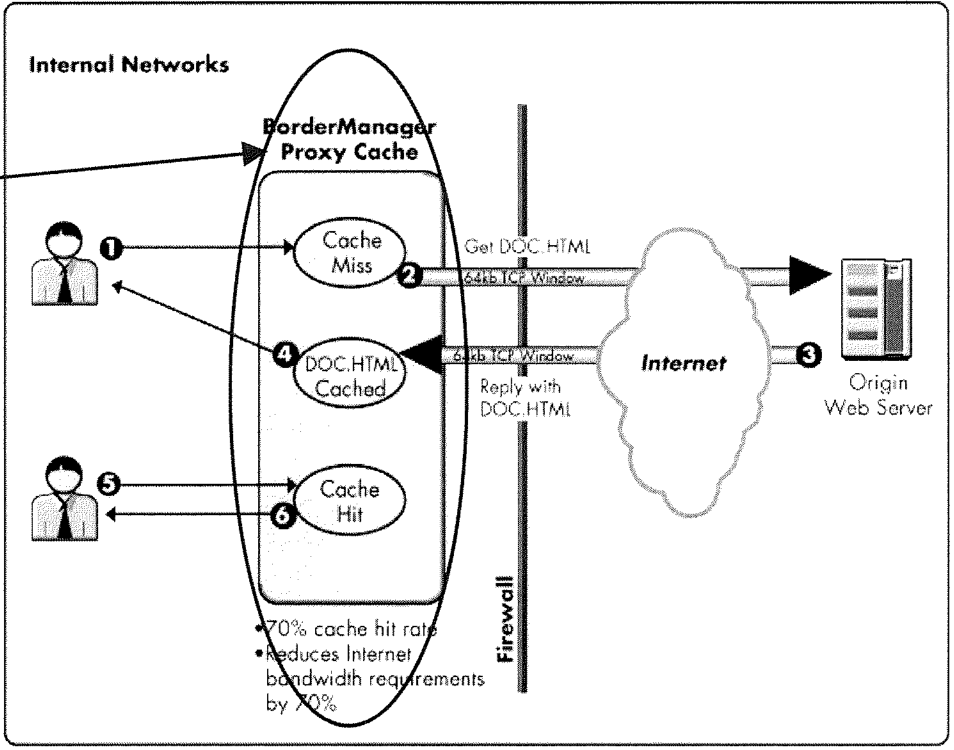
(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there;

(c) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, **the NDC of this NDC site transmitting a request for data** from this NDC site **downstream to another NDC site** closer to **the NDC server terminator site** for the stored dataset than the present NDC site;

How Proxy Cache Works

Figure 5 illustrates how BorderManager caches HTML documents and other cacheable content.

***Figure 5:** A proxy cache saves repeatedly-used objects to speed access and reduce Internet traffic.*



1. A browser issues a request for a file named DOC.HTML. This request is sent to the proxy cache over a 10 Mbps Ethernet LAN segment. In this case, the request results in a "cache miss" because the proxy cache has never serviced a request for that document before.
2. The proxy cache initiates a request for DOC.HTML from the origin web server on behalf of the browser. This request is sent over a T1 line to an ISP, then traverses the Internet until it arrives at the origin server.
3. The origin web server responds to the proxy's request by sending DOC.HTML. This transmission is much faster than a response to a browser due to the proxy's optimized receive window that can receive up to 64KB at one time and stays open to receive multiple responses. The proxy then places DOC.HTML in its cache.
4. The proxy cache responds to the original browser request with DOC.HTML.
5. Now when the same browser (or any other browser) issues a request for DOC.HTML, the request results in a "cache hit" because the proxy has kept a copy of the document in its cache.
6. In this case, the proxy replies immediately to the browser request because it has DOC.HTML in cache. The proxy's response is transmitted at 10 Mbps to the browser, eliminating the need to fetch the document again from the origin server on the Internet.

(a) **the NDC receiving the request** to access data in the stored dataset;

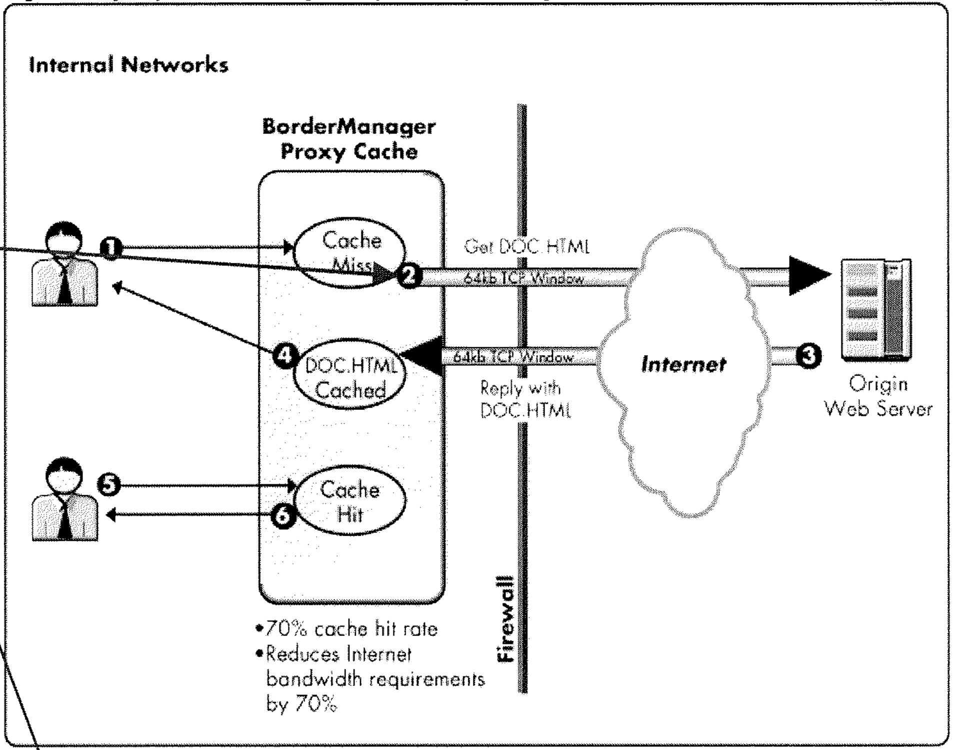
(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there;

(c) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, the NDC of this NDC site transmitting a request for data from this NDC site **downstream to another NDC site** closer to the **NDC server terminator site** for the stored dataset than the present NDC site;

How Proxy Cache Works

Figure 5 illustrates how BorderManager caches HTML documents and other cacheable content.

Figure 5: A proxy cache saves repeatedly-used objects to speed access and reduce Internet traffic.



1. A browser issues a request for a file named DOC.HTML. This request is sent to the proxy cache over a 10 Mbps Ethernet LAN segment. In this case, the request results in a "cache miss" because the proxy cache has never serviced a request for that document before.
2. The proxy cache initiates a request for DOC.HTML from the origin web server on behalf of the browser. This request is sent over a T1 line to an ISP, then traverses the Internet until it arrives at the origin server.
3. The origin web server responds to the proxy's request by sending DOC.HTML. This transmission is much faster than a response to a browser due to the proxy's optimized receive window that can receive up to 64KB at one time and stays open to receive multiple responses. The proxy then places DOC.HTML in its cache.
4. The proxy cache responds to the original browser request with DOC.HTML.
5. Now when the same browser (or any other browser) issues a request for DOC.HTML, the request results in a "cache hit" because the proxy has kept a copy of the document in its cache.
6. In this case, the proxy replies immediately to the browser request because it has DOC.HTML in cache. The proxy's response is transmitted at 10 Mbps to the browser, eliminating the need to fetch the document again from the origin server on the Internet.

(a) **the NDC receiving the request** to access data in the stored dataset;

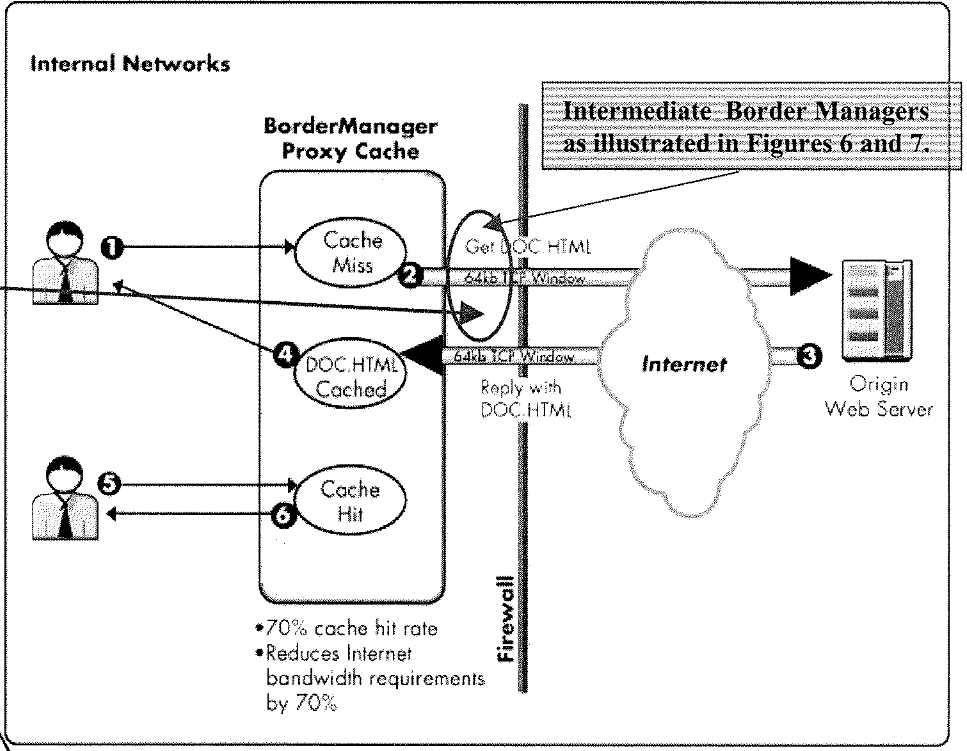
(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there;

(c) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, **the NDC of this NDC site transmitting a request for data** from this NDC site downstream to another NDC site closer to **the NDC server terminator site** for the stored dataset than the present NDC site;

How Proxy Cache Works

Figure 5 illustrates how BorderManager caches HTML documents and other cacheable content.

Figure 5: *A proxy cache saves repeatedly-used objects to speed access and reduce Internet traffic.*



1. A browser issues a request for a file named DOC.HTML. This request is sent to the proxy cache over a 10 Mbps Ethernet LAN segment. In this case, the request results in a "cache miss" because the proxy cache has never serviced a request for that document before.
2. The proxy cache initiates a request for DOC.HTML from the origin web server on behalf of the browser. This request is sent over a T1 line to an ISP, then traverses the Internet until it arrives at the origin server.
3. The origin web server responds to the proxy's request by sending DOC.HTML. This transmission is much faster than a response to a browser due to the proxy's optimized receive window that can receive up to 64KB at one time and stays open to receive multiple responses. The proxy then places DOC.HTML in its cache.
4. The proxy cache responds to the original browser request with DOC.HTML.
5. Now when the same browser (or any other browser) issues a request for DOC.HTML, the request results in a "cache hit" because the proxy has kept a copy of the document in its cache.
6. In this case, the proxy replies immediately to the browser request because it has DOC.HTML in cache. The proxy's response is transmitted at 10 Mbps to the browser, eliminating the need to fetch the document again from the origin server on the Internet.

(a) **the NDC receiving the request** to access data in the stored dataset; ✓

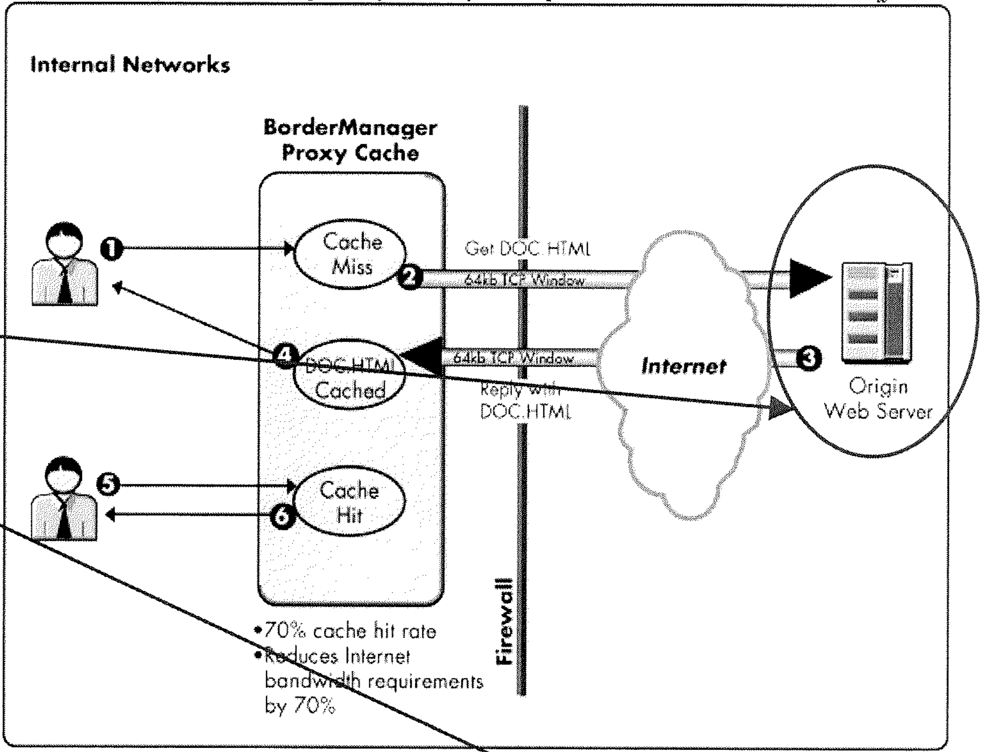
(b) the NDC **checking the NDC buffer** at this NDC site to determine if **a projected image of data requested** from the stored dataset is already present there; ✓

(c) if **the NDC buffer of this NDC site does not contain a projected image of all data requested** from the stored dataset, and if the NDC site receiving the request **is not the NDC server terminator site** for the stored dataset, **the NDC of this NDC site transmitting a request for data** from this NDC site **downstream to another NDC site** closer to the **NDC server terminator site** for the stored dataset than the present NDC site;

How Proxy Cache Works

Figure 5 illustrates how BorderManager caches HTML documents and other cacheable content.

Figure 5: *A proxy cache saves repeatedly-used objects to speed access and reduce Internet traffic.*



1. A browser issues a request for a file named DOC.HTML. This request is sent to the proxy cache over a 10 Mbps Ethernet LAN segment. In this case, the request results in a "cache miss" because the proxy cache has never serviced a request for that document before.
2. The proxy cache initiates a request for DOC.HTML from the origin web server on behalf of the browser. This request is sent over a T1 line to an ISP, then traverses the Internet until it arrives at the origin server.
3. The origin web server responds to the proxy's request by sending DOC.HTML. This transmission is much faster than a response to a browser due to the proxy's optimized receive window that can receive up to 64KB at one time and stays open to receive multiple responses. The proxy then places DOC.HTML in its cache.
4. The proxy cache responds to the original browser request with DOC.HTML.
5. Now when the same browser (or any other browser) issues a request for DOC.HTML, the request results in a "cache hit" because the proxy has kept a copy of the document in its cache.
6. In this case, the proxy replies immediately to the browser request because it has DOC.HTML in cache. The proxy's response is transmitted at 10 Mbps to the browser, eliminating the need to fetch the document again from the origin server on the Internet.